PalmSpace: Continuous Around-Device Gestures vs. Multitouch for 3D Rotation Tasks on Mobile Devices

Sven Kratz, Michael Rohs
University of Munich
Amalienstr. 17
80333 Munich, Germany
{firstname.lastname}@ifi.lmu.de

Dennis Guse*, Jörg Müller†,
Gilles Bailly‡, Michael Nischt§
Telekom Innovation Laboratories
TU Berlin, Ernst-Reuter-Platz 7
10587 Berlin, Germany

ABSTRACT

Rotating 3D objects is a difficult task on mobile devices, because the task requires 3 degrees of freedom and (multi-)touch input only allows for an indirect mapping. We propose a novel style of mobile interaction based on mid-air gestures in proximity of the device to increase the number of DOFs and alleviate the limitations of touch interaction with mobile devices. While one hand holds the device, the other hand performs mid-air gestures in proximity of the device to control 3D objects on the mobile device's screen. A flat hand pose defines a virtual surface which we refer to as the *PalmSpace* for precise and intuitive 3D rotations. We constructed several hardware prototypes to test our interface and to simulate possible future mobile devices equipped with depth cameras. We conducted a user study to compare 3D rotation tasks using the most promising two designs for the hand location during interaction – behind and beside the device – with the virtual trackball, which is the current state-of-art technique for orientation manipulation on touchscreens. Our results show that both variants of PalmSpace have significantly lower task completion times in comparison to the virtual trackball.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Input devices and strategies, Interaction Styles

General Terms

Human Factors, Design, Experimentation

*dennis.guse@telekom.de
†joerg.mueller@tu-berlin.de
‡gillesbailly1@googlemail.com
§michael.nischt@tu-berlin.de

AVI '12 May 21-25, 2012, Capri Island, Italy

Figure 1: Using the pose of the flat hand behind the device to freely rotate a 3D object. A depth camera is used to determine the hand posture.

Keywords

Around-device interaction, depth camera, mobile interaction, 3D user interfaces, 3D Rotation, input devices

1. INTRODUCTION

Current graphics hardware for mobile devices now allows for rendering sophisticated 3D scenes on mobile devices. This capability is useful for gaming, but also allows implementing CAD tools and other scenarios, such placement of virtual furniture in a mobile AR environment or browsing an online shopping catalogue in 3D. While it is now possible to deliver useful 3D content on mobile devices, interacting with this type of content is still challenging. Users not only face the limitations of direct finger interaction on touchscreens caused by the fat finger problem, i.e. occlusion and accuracy [19], but 3D content itself requires more than the two degrees of freedom generally provided by touchscreens.

In this paper, we focus on one of the two fundamental 3D operations: the rotation within 3D scenes or of objects, which is not adequately supported on current mobile devices. The current solutions for use on (multi-)touchscreens only allow for an indirect mapping of 2D input gestures to 3D object rotation. For instance, the virtual trackball metaphor [7] provides an invisible sphere on top of the object. A 2D finger movement on this sphere rotates the 3D object. However, this technique has the drawback that the movement is mapped indirectly to rotation.

To overcome the limitations of touchscreens, we propose to interact around the device for manipulating 3D content. Around-device interaction [12] allows separating the input and the output of the mobile device by performing gestures in *proximity* of the device. We extend this concept by enabling the space around the device for gestural interaction:

The gesture space is now delineated by the reach of the user's arm. We denote this reachable space as *PalmSpace*. This is well suited for 3D interaction, because 1) it increases the degrees of freedom for input, 2) it provides a finer control as a larger interaction volume is available and 3) it allows more natural interaction as 3D operations can be directly mapped to 3D gestures. Using the space behind and beside the device for gestural interaction has the advantage of being occlusion-free, as the hand does not cover the device's display, and provide a large input space. We refer to the spaces behind and next to the device as *BackSpace* and *SideSpace*, respectively.

We also propose a novel gesture for performing 3D rotations called *Palm Gesture*. A user holds the device in one hand and uses the non-holding hand to perform the gesture in the PalmSpace. The user orients the palm of the hand, which defines a plane, to manipulate the orientation of the 3D object/scene on the screen. This has the advantage to introduce a direct mapping between the gesture and the 3D operations, which is easy to understand and easy to learn for novice users and efficient for expert users.

We argue that such interfaces can be facilitated with depth cameras, which provide depth information for each pixel. We present a proof-of-concept based on a depth camera attached to an iPhone to capture the palm posture (Figure 1). It is reasonable to assume that manufacturers will be able to equip mobile devices with depth-sensing cameras in the future. In fact, some manufacturers already equip mobile phone with stereo RGB cameras. [1]

We conducted a user study to determine the preferred hand poses for around-device gestural interaction. Based on these results of we report the findings of a user study comparing 3D rotation using SideSpace and BackSpace with the virtual trackball [7] as a baseline. The results show that both SideSpace and BackSpace have lower task execution times and obtained ISO9241-9 ratings that are similar to the virtual trackball. .

The remainder of the paper is structured as follows: we present related work, detail the main underlying concepts of the PalmSpace approach, describe our prototype implementation, report the results of a user study, and draw implications for future around-device interfaces.

2. RELATED WORK

2.1 Interaction on Touch Screens

The interface of mobile devices is impoverished in comparison with standard computers. Users directly interact with fingers making precise selection difficult due to the large contact area between the fingertip and the touchscreen. Moreover, the finger may occlude large parts of the screen. These issues are known as the "fat finger problem" [19]. Finally, the amount of screen real estate is severely restricted. These three limitations make pointing tasks [21] or interaction with traditional widgets (for interacting with 3D objects) difficult. Interaction on the rear of the device [22] has the advantage of avoiding occlusion on small touchscreens. For instance, LucidTouch [22] adds a touch-sensitive surface to the rear of the device and also allows the device to sense a hover state above the rear touch-sensitive surface. However, the technique does not allow 3D gestures and the input space is still limited.

2.2 Interaction Around the Device

An alternative consits in performing around-the-device interaction [2, 6, 10, 12]. For instance, SideSight [2] uses IR distance sensors to implement multi-touch input in the area on the sides of the device, but only works when the device is placed on a flat surface. In a similar way, HoverFlow [12] allows simple hand gesture recognition above the device's screen, also using IR distance sensors. Harrison et al. [6] and Ketabdar et al. [10] implemented around device interaction by detecting a magnetic token that is worn on the user's hand with a magnetic field sensor. Such a sensor is present in many current mobile devices. However, it is difficult to derive precise 3D position information from these sensors and the technique forces users to have a magnet on the finger. All these techniques are based on an interaction in very close proximity of the device with a couple of centimeters maximum distance. In contrast, PalmSpace allows users to interact in a larger input space (arm's reach) and focus on interaction with 3D content as position and pose are recognized.

2.3 Manipulating 3D Content

Mobile interaction with 3D content is often implemented using the device's accelerometer and magnetometer (compass). These sensors provide a reference orientation for the 3D objects, i.e. for augmented reality browser applications such as Layar[2]. The advantage of using this sensor combination is that the 3D scene can be viewed and controlled by holding the device with a single hand. Nevertheless, a problem when interacting using this technique is that the orientation of the display may not always be optimal for viewing, due to the tilt required to control the application. A study by Kratz et al. [13] has shown that using a virtual trackball on a touchscreen for 3D rotation tasks on a mobile device outperforms tilt-based control approaches. Hachet et al. [5] use a regular camera to detect color codes on a piece of cardboard for controlling a 3D object. This approach requires markers in the camera view and does not allow for rapid change between different gestures.

2.4 Interaction with Depth Sensors

PalmSpace is also related to interaction with depth cameras and hand posture recognition [1, 11]. For instance, Kollorz et al. [11] proposed an algorithm for static hand posture recognition using a depth camera. These works mostly focus on *symbolic* recognition of certain hand postures or gestures, whereas this work emphasizes the use of the hand as a natural and *continuous* control mechanism for 3D content in mobile applications. Finally, Bailly et al. recently proposed ShoeSense [1], a system with a depth sensor located on the shoe. Users can perform hand gestures in the air to control their mobile devices. While this location makes gesture recognition easier [1], it forces users to have an additional device on the shoe.

Expanding upon the concepts just described, PalmSpace uses the 3D space around the device for hand and finger

[1]e.g. the LG Optimus 3D:
`http://www.lg.com/uk/mobile-phones/all-lg-phones/LG-android-mobile-phone-P920.jsp`

[2]`http://layar.com`

gestures without the need for additional tokens. To understand the ergonomic limitations of gestural interaction in this space, we implemented a 3D viewer application. It is based on a "palm plane" metaphor to control 3D object rotation (Figure 1).

3. THE PALMSPACE

We refer to the 3D space around the device that allows manipulating 3D virtual objects via hand gestures as the PalmSpace. One of the advantages of this approach is that the 3D space behind and to the side of the device avoids display occlusion and also achieves a close spatial correspondence between virtual 3D objects and movements in the physical space around the screen. However, we also found a number of critical aspects when designing gestures for this space by analyzing (1) the ergonomic properties of the interaction volume, (2) the types of gestures that can be detected robustly using a depth camera and (3) possible ways of mapping the data obtained from the depth camera to the virtual object control. We now detail these three aspects.

3.1 Interaction Volume

The interaction volume is bound by the pyramid-shaped camera's viewing frustum, i.e. its angle of view and depth range. The corresponding interaction volume is further restricted by the arm's reach of the user. Furthermore, entering and exiting the volume can be used as an implicit clutching or "action selection" mechanism[3]).

3.2 Gesture Types

PalmSpace allows for the recognition of a wide variety of gesture types. In this paper, we chose to implement rotation gestures as a proof of concept that demonstrates PalmSpace interaction. More complex gestures and input techniques, such as pointing or palm-bending can for instance be implemented using machine learning techniques such as skeletal matching [16].

Nevertheless, there are two types of limitations which need to be taken into account for PalmSpace interaction. The first is that the depth-map image provided by the camera scans the 3D volume from one perspective only and hence occlusions can occur. This becomes an issue if multiple fingers have to be tracked and one finger is hiding another while the hand is rotating. Therefore, we focus on palm gestures in which this kind of occlusion cannot occur. Secondly, there are ergonomic constraints. In particular, rotations around the wrist are only possible within certain limits and may be uncomfortable. Moreover, precision becomes an issue at the limits of the wrist's rotation range [17].

With the above in mind, we propose the *flat hand* as a useful continuous gesture, which we implemented for the prototype and evaluated in the user study. More precisely, the palm approximates a plane, which yields two parameters: *direction* orthogonal to the plane (surface normal) and *distance* to the camera center (origin of the coordinate system).

Other types of hand gestures are to be explored in the future – potentially with improved hardware – include finger position and direction, palm bending, or general finger gestures. This opens a large space of potential high-dexterity gestures, however limitations of the depth camera have to be considered in the design of suitable gesture sets.

[3]This has already been implemented for RGB cameras and is available as a commercial library at `http://www.eyesight-tech.com`

3.3 Gesture Parameter Mapping

The presented gesture needs to be mapped to the orientation of the manipulated virtual object. There are several options to accomplish this. The first option is to link the hand orientation to the virtual object, so users can change the orientation of the object by rotating their hand. This corresponds to *absolute control*, in which the hand orientation defines the orientation of the controlled object. This mapping is very intuitive but limited, due to the rotational ergonomic constraints of the hand and thus does no allow complete, 360°, rotations.

A variant of this mapping is *scaled absolute control*, which extends absolute control by a rotational scaling factor. This enables full rotation of the object with a smaller rotation of the hand, but it is likely that scaling will degrade precision.

Finally, *rate control* maps the angle between the initial and the current hand pose to the rotational speed of the virtual object. This allows to rotate the object completely, but it needs to be done over time rather than defining the orientation directly and is prone to overshoot. In fact, Oakley and O'Modhrain [15] found that rate-control is significantly slower than absolute control.

As an alternative, the concept of absolute control can be extended using clutching. Here, the hand orientation is not used to define the orientation of the virtual object absolutely, but instead relative to the orientation of the object and the hand at the time the clutching was engaged. This is called *relative control*. It allows performing complete rotation under the trade-off to use multiple gestures and thus degrade performance, but preserves the intuitive mapping as the virtual object follows the rotation of the hand once the clutch is engaged. Engaging and disengaging the clutch can be accomplished by moving the hand in and out of the depth camera's viewing frustrum.

4. PALMSPACE PROTOTYPE

We built a prototype system to evaluate PalmSpace interaction. The prototype consists of a depth camera mounted on a iPhone (Figure 2), a mobile application and a PC that performs computer vision to extract control inputs from the depth images for use by the mobile application.

4.1 BackSpace and SideSpace Prototype

For depth imaging, we use a Mesa Imaging Swiss Ranger 4000 camera (SR4K)[4], which uses modulated IR light to determine the pixel depth values. Under optimal conditions the camera achieves an absolute accuracy (deviation of the mean measured distance from actual distance) of less than ±10 mm and a repeatability (standard deviation of measured distances) of 4-7 mm. The camera has a resolution of 176 × 144 pixels and a horizontal and vertical field of view (FOV) of 44° and 35°, respectively. For close and/or highly reflective objects, the image can become partially saturated and provide no reliable depth information. The camera is optimized for distances from 60 cm to 4.4 m, which is too far for the intended PalmSpace setup. Using a low integration time of 1.8 ms and disabling some IR emitters using adhe-

[4]`http://mesa-imaging.ch`

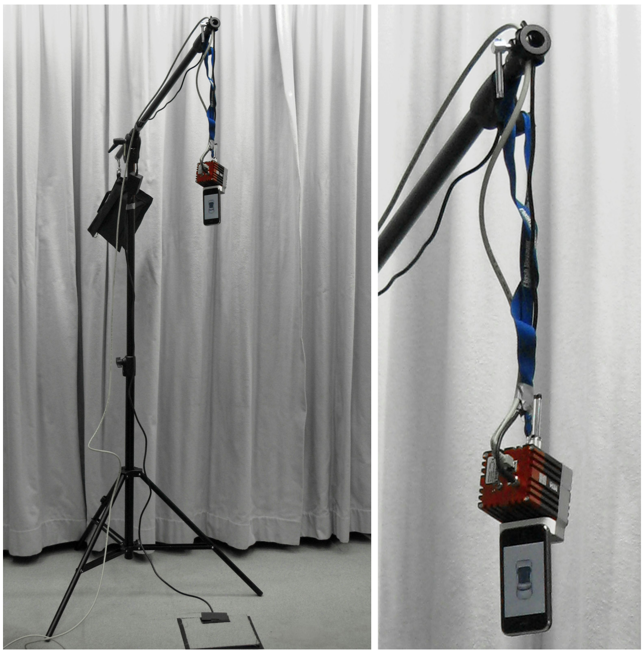


Figure 2: Hardware setup: The depth camera is mounted on top of a mobile phone in a slightly downward angle. The prototype is attached to a lanyard, such that it can be handled comfortably.

sive tape, we decreased the minimal distance from hand to camera down to 15 cm at the cost of accuracy. At the minimum distance the interactive area has a width of 12 cm and a height of 9.6 cm.

The complete setup aims at mediating the currently bulky hardware and simulating a system that is completely integrated into the mobile phone. For the BackSpace prototype the camera is attached above an iPhone 4 (upside down, portrait orientation) via an iPhone dock in a 38° downward looking angle (Figure 2). The SideSpace prototype consists of the iPhone flexibly suspended from the top in landscape orientation, for easier handling, and the depth camera attached to the backside using adhesive tape. We had to use different device orientations for BackSpace and SideSpace due to mechanical reasons. For SideSpace we adjusted the order of the rotation axes, so that the mapping from hand pose to object rotation remained identical to BackSpace. In order to relieve the user from the need to carry the additional weight of the camera, 510g, the prototype is attached using a lanyard to a Manfrotto 420b lighting tripod. In this way, the prototype hangs freely in front of the users and they are not required to hold the relatively heavy setup.

The described setup allows the iPhone to be viewed and handled comfortably, while the non-holding hand is free for interaction behind or beside the prototype, respectively, in the FOV of the camera.

4.2 Gesture Recognition Application

For the vision processing, we use a Dual-Xeon Quadcore 2.66GHz CPUs running Linux. The prototypical gesture recognition application is written in C++ using the Swiss-Ranger Linux Driver[5] and the Point Cloud Library (PCL)[18].

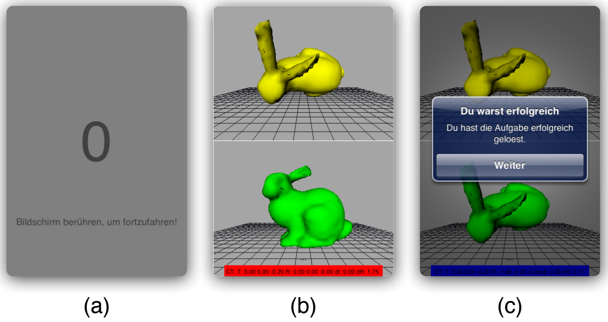


Figure 3: The sequence of screens shown for each trial in the user study. (a) Countdown screen that can be dismissed with a tap. (b) Trial screen that shows the target object on top and the controllable object at the bottom. (c) Feedback screen that indicates success or failure of a trial.

The RANSAC[3] algorithm is used to estimate the rotational parameters of the plane described by the user's flat hand on a downsampled image, which is segmented using a depth threshold. The application estimates only the rotation around the x- and y-axes as the z-axis could not be reliably estimated using the camera setup. The estimated Euler angles are transferred via UDP multicast to the iPhone using a wireless router. On the hardware we are using the application runs at 15 Hz.

4.3 Mobile Application

We developed an iPhone 4 application as a demo application for the interaction concept and as a test application for the user study. The application allows rotating the Stanford Bunny [20] in 3D using either a conventional touchscreen approach or PalmSpace gestures. Rotation via the touchscreen is realized by a virtual trackball [7]. The trackball allows rotation on all three axes by mapping the rotation input to a 3D Gaussian bell curve and converting the 3D translation on the curve to a rotation relative to the Gaussian's center as an angle/axis pair [13]. For PalmSpace the mobile application uses the Euler angles produced by the gesture recognition application (see previous paragraph) to calculate a rotation matrix for the Stanford Bunny.

Figure 3 shows the screens of the application: countdown (a), trial (b) and feedback (c). The countdown screen can be dismissed by tapping the screen. The top/right half of the trial screen shows the target orientation of the object. The bottom/left half of the screen shows the user-controllable object.

When the demo application is used for the user study, trials are served to the mobile device via HTTP. Each trial is separated by a 5 second countdown screen (Figure 3 (a)). When the countdown has reached zero, the countdown screen can be dismissed by tapping on the device's screen, and the new trial begins. Once a trial has completed, the device reports either success or failure of the task (Figure 3 (c)) and the task duration to the server.

5. USER STUDY

In our user study, we compared BackSpace, SideSpace and a touch-based virtual trackball for 3D rotation tasks.The

[5]The driver is available at http://mesa-imaging.ch/drivers.php

goal of the study was to show that BackSpace and SideSpace outperform the virtual trackball and are also rated higher, mainly because of the direct correspondence between palm rotation in 3D space on virtual object rotation.

5.1 Pilot Study

Initially, the optimal placement of the hand relative to the device was unclear to us. Within the design space we are using, we identified three useful hand poses: hand in front of the device (p_{front}), hand to the side of the device (p_{side}) and the hand behind the device (p_{back}). Figure 4 illustrates the proposed hand poses. To determine the preferred and most comfortable pose, we conducted a pilot study with five participants. We asked them to simulate interaction in positions p_{front}, p_{side}, p_{back} as well as with the virtual trackball ($p_{\text{trackball}}$) as a baseline technique. After the participants made clear that they understood the interaction metaphor and the task to be accomplished, we asked them to rate their comfort levels following ISO 9241-9, and to provide a ranking of the techniques p_{front}, p_{side}, p_{back}, and $p_{\text{trackball}}$. The results of the pilot study indicate that p_{side} is the preferred gesture-based technique. It was rated better than p_{front} and p_{back} for comfort and was unanimously ranked higher than the competing gesture-based techniques. Probably owing to familiarity to touchscreen-based interaction, $p_{\text{trackball}}$ was rated best for comfort and was placed first in the ranking. The results of the main study, however, demonstrate the advantages of the gesture-based technique. Because hand pose p_{front} was rated lowest amongst the gesture-based techniques and because it leads to occlusion of the screen contents, we decided to drop this technique in the main experiment.

5.2 Hypotheses

Having obtained insights into the preferred pose of the hand relative to the device, we formulated the following hypotheses for our experiment:

- **H1** *BackSpace (p_{back}) and SideSpace (p_{side}) have lower task completion times than the virtual trackball (p_{front}).*
We presume that BackSpace and SideSpace achieve a lower overall task completion time as those techniques provide a direct mapping.

- **H2** *SideSpace (p_{side}) has a lower overall task completion time than BackSpace.*
We presume that SideSpace achieves a lower overall task completion time than BackSpace as the results of the pilot study revealed the hand pose of SideSpace is perceived as more satisfying.

- **H3** *BackSpace (p_{back}) and SideSpace (p_{side}) are rated lower with regard to the required force and the overall effort than the virtual trackball ($p_{\text{trackball}}$).*
We presume that the PalmSpace techniques are rated lower with regard to the required force and the overall effort as the virtual trackball only requires small finger movements whereas the PalmSpace techniques require to hold and move the hand.

5.3 Design of Experiment

In the main experiment, we compared touch ($p_{\text{trackball}}$) with our gestural rotation techniques using poses p_{side} and p_{back}. The experiment used a counterbalanced within participants design with interaction technique as the single factor.

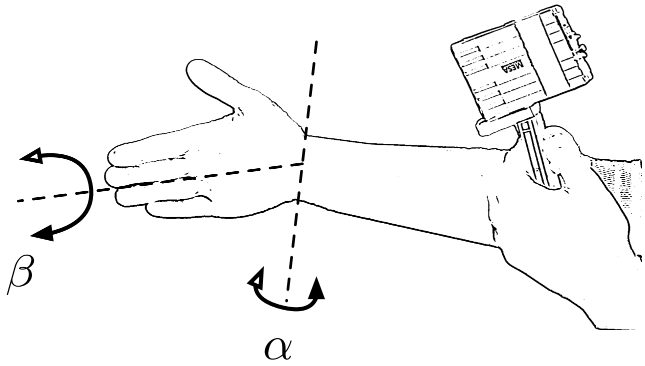


Figure 5: The Euler angle α is controlled by wrist flexion, whereas β is controlled using pronation and supination.

The levels of this factor are SideSpace (p_{side}), BackSpace (p_{back}), and virtual trackball ($p_{\text{trackball}}$).

5.3.1 Participants and Apparatus

We invited 5 male and 5 female participants (mean age 27, age range 22-47), all of them were right-handed, and were compensated with a voucher for their participation. None of them participated in the pilot study. All participants were experienced in touchscreen-based interaction with mobile devices. For the BackSpace and SideSpace gestures the participants used the mobile application and setups described in Section 4. For the virtual trackball interaction they used the same application, but the unmodified iPhone 4 only (not suspended to the ceiling).

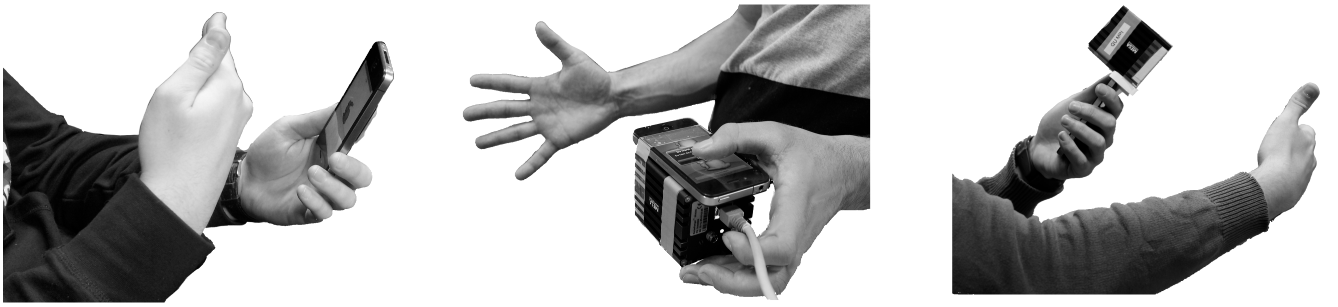
5.3.2 Procedure and Tasks

The user's task was to rotationally align a 3D-object with a presented target as fast as possible. The control object and the target object were both Stanford Bunnies. The user interface for the rotation task is shown in Figure 3 (b). For each technique the user had to perform 24 trials. The trials consisted of single and combined rotations around the x-axis and y-axis with the Euler angles α and β. Before conducting the trials with each interaction technique the participants had approximately 5 minutes time to try out and explore the technique.

In terms of human physiology [4, 14], the Euler angle α corresponds to the flexion rotation of the human wrist, and β corresponds to both pronation and supination. The hand movements required to rotate around α and β are further depicted in Figure 5. Keeping within the detection capabilities of our gesture-based system and the ergonomic constraints of wrist rotation, we selected the following values in degrees for α and β for use in the trials:

$$\alpha \in \{-50, -40, -35, -30, -25, -15, 0, 15, 20, 30, 50\}$$
$$\beta \in \{-70, -50, -40, -25, -20, -15, 0, 10\}$$

For a trial to be completed successfully, the user had to dwell for 300ms within a rotation distance [8] of $\Phi_\epsilon = 9.9°$, which corresponds to a maximal offset of ± 7 degrees for α and β, respectively. The dwell time is included in the completion times reported in the results section. If, after 30s the user had failed to achieve a rotation satisfying these criteria, the trial was aborted and marked as unsuccessful.



(a) (b) (c)

Figure 4: The hand poses we considered in the pre-study: (a) in front of the device (p_{front}), (b) beside the device (*SideSpace*, p_{side}) and (c) behind the device (*BackSpace*, p_{back}).

5.3.3 Measured Variables

To evaluate the performance of BackSpace, SideSpace, and virtual trackball for the rotational task, we measured the time to completion for each trial and the number of trials that could not be completed. In addition, the users had to fill out a questionnaire after completing all trials for a given technique. The questionnaire is based on ISO9241-9 [9]. At the end of the experiment the user was asked to fill out an additional questionnaire, which asked to order the interaction techniques according to their personal preference, to rate the intuitiveness and to denote the most comfortable posture. Finally, the users had to indicate which input technique they would be comfortable with using in public places.

5.4 Results

In the following we present our observation on the interaction postures shown by the test subjects during the experiment, as well as the results for task completion times and the user questionnaires.

5.4.1 Interaction Posture

BackSpace (p_{back}) and SideSpace (p_{side}) require two handed interaction as one hand needs to hold the mobile device and the non-holding hand is used for interaction. This posture was explained and shown to the participants. Interaction with the virtual trackball ($p_{\text{trackball}}$) is possible one-handed using the thumb on the touchscreen as well as using two hands, i.e. using one hand to hold the device and one finger of the non-holding hand for interaction. For this interaction technique the participants were neither instructed nor shown if interaction should be done one- or two-handed. In the exploration phase before the test trials, two participants first explored one-handed interaction, but switched to two handed interaction as they found the thumb to be too imprecise as well as to big with regard to occlusion. All other participants directly used two-handed interaction. All participants solved the virtual trackball trials using two hands.

5.4.2 Task Completion Times

We recorded a total of 240 trials for each technique. Not all trials were completed successfully. For BackSpace (p_{back}) 20 trials (8.3%), for SideSpace (p_{side}) 21 (8.8%) and for the virtual trackball ($p_{\text{trackball}}$) 22 (9.2%) trials were marked as being unsuccessful.

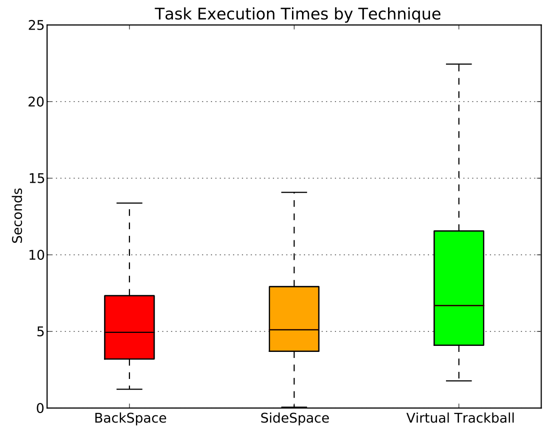


Figure 6: Box plots of the task completion times by input technique.

Of the successful trials, we obtained the following average task completion times for BackSpace (p_{back}) 6.09s, $\sigma = 4.36$, for SideSpace (p_{side}) 7.19s, $\sigma = 5.6$ and for the virtual trackball ($p_{\text{trackball}}$) 8.45s, $\sigma = 5.82$. A box plot of the results for task completion time is shown in Figure 6.

A histogram analysis of the task completion data showed a strong left skew of the task completion times. We log-transformed the data to obtain an unskewed Gaussian distribution of the data. We conducted an ANOVA on the log-transformed task completion data and found a significant effect for input technique: $F(2, 16) = 10.42$; $p < 0.001$. A Sidak post-hoc analysis indicates that the differences between each of the techniques are significant:

- BackSpace (p_{back}) vs. SideSpace (p_{side}): $p = 0.04$
- BackSpace (p_{back}) vs. virtual trackball ($p_{\text{trackball}}$): $p < 0.001$
- SideSpace (p_{side}) vs. virtual trackball ($p_{\text{trackball}}$): $p = 0.037$

Thus, **H1** is validated as BackSpace and SideSpace show lower average task completion times than the virtual trackball. However **H2** could not be confirmed as BackSpace

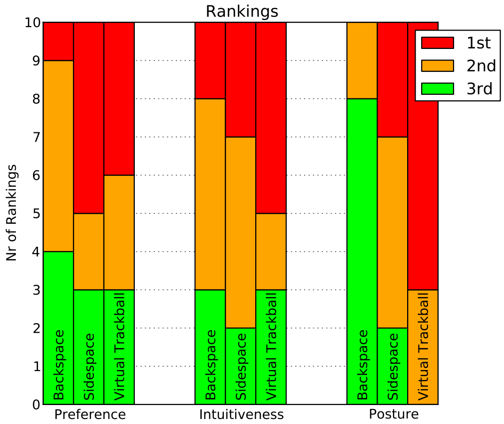


Figure 7: This Figure shows the number of rankings given by input technique for *preference*, *intuitiveness* and *posture*.

shows significantly lower average task completion times than SideSpace, even though SideSpace is the preferred input technique of the pilot study.

5.4.3 User Evaluation

In the final questionnaire the participants were asked to order the interaction techniques according to their personal preference and their perceived intuitiveness. Figure 7 shows the results. Employing the Kruskal-Wallis h-test, we could neither find a statistically significant mean difference for personal preference ($\chi^2 = .817; p = 0.665$), nor for intuitiveness ($\chi^2 = 2,030; p = 0.362$).

After completing all trials for each interaction technique the participants were asked to complete the ISO9241-9 questionnaire. Figure 8 shows the results. A Kruskal-Wallis h-test shows a significant result for the overall effort ($\chi^2 = 10.225$; $p = 0.006$) as well as the required force ($\chi^2 = 10.205$; $p = 0.006$). Thus, **H3** is validated as the user perception of required force and overall effort of SideSpace and BackSpace are rated significantly lower than the virtual trackball.

In public situations, 6 of the 10 participants would use BackSpace and the remainder would use SideSpace. In terms of qualitative ratings, the virtual trackball outperforms both PalmSpace variants as all participants would use the virtual trackball in public places. However, it must to be kept in mind that the participants encountered the PalmSpace interface the first time whereas touchscreen-based interaction is widely known and adopted.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented two new interaction techniques for manipulating the orientation of virtual objects on mobile device screens. We introduced the “palm metaphor”, which makes manipulation intuitive and fast as the manipulated object always adopts the same orientation as the palm. The results of the presented user study show that our prototypical implementation performs very well with regard to performance as well as user preference and intuitiveness. Most notably, the two presented variants of PalmSpace perform better than the virtual trackball, which is the current state-of-the-art solution for rotational tasks on mobile devices using a touchscreen. Moreover, the PalmSpace can potentially be extended to translation tasks as well. Hence it is reasonable to assume that controlling 6DOF is achievable using the palm metaphor, which would make it fully suited for viewing and manipulating 3D objects as well as navigating in 3D environments like in navigational tasks or gaming.

The results we have obtained in this work imply that around-device interfaces for mobile devices are likely to improve the usability for spatial tasks requiring a high number of simultaneous degrees of freedom as well as high precision. We believe that the direct manipulation metaphor afforded by SideSpace allowed the users to rotate the 3D object more intuitively than the virtual trackball because

7. REFERENCES

[1] G. Bailly, J. Müeller, M. Rohs, D. Wigdor, and S. Kratz. Shoesense: Shoesense: A new perspective on hand gestures and wearable applications. In *Proceedings of the SIGCHI conference on human factors in computing systems (CHI 2012)*. ACM, 2012.

[2] A. Butler, S. Izadi, and S. Hodges. SideSight: Multi-touch interaction around small devices. In *Proc. UIST*, pages 201–204, 2008.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[4] E. Grandjean. *Fitting the task to the man: A textbook of occupational ergonomics.* Taylor & Francis/Hemisphere, 1989.

[5] M. Hachet, J. Pouderoux, and P. Guitton. A camera-based interface for interaction with mobile handheld computers. In *Proc. Symp. Interactive 3D Graphics and Games*, pages 65–72, 2005.

[6] C. Harrison and S. E. Hudson. Abracadabra: Wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proc. UIST*, pages 121–124, 2009.

[7] K. Henriksen, J. Sporring, and K. Hornbæk. Virtual trackballs revisited. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):206–216, 2004.

[8] D. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[9] ISO. Ergonomic requirements for office work with visual display terminals (vdts) - part 9: Requirements for non-keyboard input devices. In *International Organization for Standardization, ISO/DIS 9241-9*, 2000.

[10] H. Ketabdar, K. Yüksel, and M. Roshandel. MagiTact: Interaction with mobile devices based on compass (magnetic) sensor. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 413–414. ACM, 2010.

[11] E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a time-of-flight camera. *Int. J. of Intelligent Systems Technologies and Applications*, 5(3):334–343, 2008.

[12] S. Kratz and M. Rohs. HoverFlow: Expanding the design space of around-device interaction. In *Proc. MobileHCI*, pages 31–38, 2009.

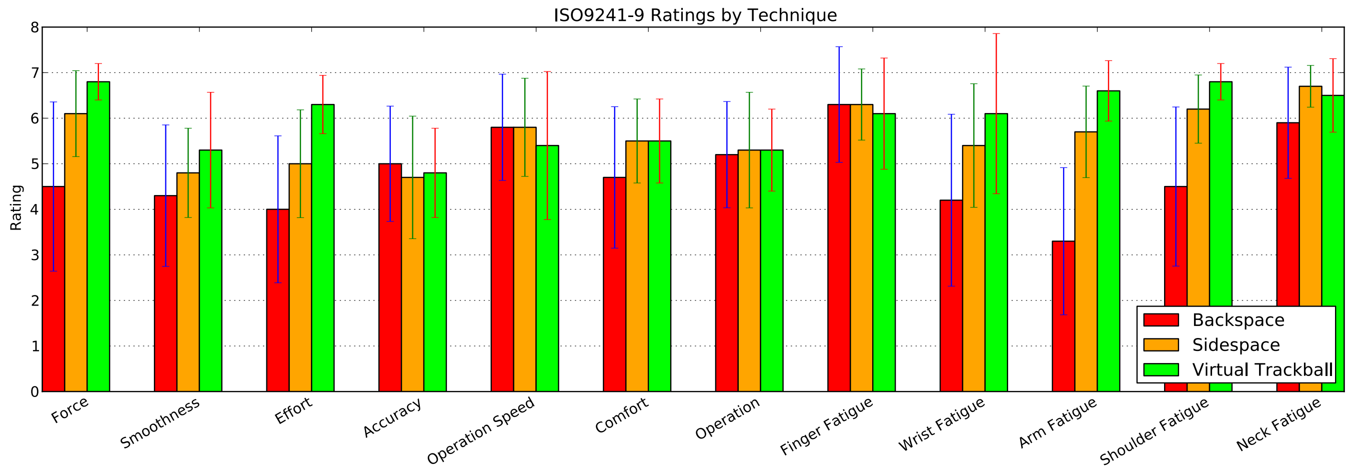


Figure 8: The average ISO90241-9 ratings given by technique on a seven point Likert scale. The error bars indicate the standard deviation.

[13] S. Kratz and M. Rohs. Extending the virtual trackball metaphor to rear touch input. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 111–114. IEEE, 2010.

[14] NASA. *Man-Systems Integration Standards*. Rev. B, 1995.

[15] I. Oakley and S. O'Modhrain. Tilt to Scroll: Evaluating a Motion Based Vibrotactile Mobile Interface. In *IEEE Proceedings of the World Haptics Conference*, Pisa, Italy, 2005.

[16] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677–695, 1997.

[17] M. Rahman, S. Gustafson, P. Irani, and S. Subramanian. Tilt techniques: Investigating the dexterity of wrist-based input. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1943–1952. ACM, 2009.

[18] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[19] K. Siek, Y. Rogers, and K. Connelly. Fat finger worries: How older and younger users physically interact with pdas. *Human-Computer Interaction-INTERACT 2005*, pages 267–280, 2005.

[20] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994.

[21] D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 657–666. ACM, 2007.

[22] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. LucidTouch: A see-through mobile device. In *Proc. UIST*, pages 269–278, 2007.